Rewarding Probabilistic Hybrid Automata

Ernst Moritz Hahn

Institute of Software Chinese Academy of Sciences

September 27th, 2013

joint work with Holger Hermanns
real-world systems: combination of digital controller and continuous environment
real-world systems: combination of digital controller and continuous environment

often safeness critical
real-world systems: combination of digital controller and continuous environment

often safeness critical

need to formally analyse such hybrid systems
“does a computing system fulfil its specification?”
Model Checking

“does a computing system fulfil its specification?”

- formal model $\mathcal{M}$ of system
Model Checking

“does a computing system fulfil its specification?”

- formal model $M$ of system
- specification $\phi$
Model Checking

“does a computing system fulfil its specification?”

- formal model $M$ of system
- specification $\phi$
- automatic proof or refutation of

$M \models \phi$
Model Checking

“does a computing system fulfil its specification?”

- formal model $\mathcal{M}$ of system
- specification $\phi$
- automatic proof or refutation of

$$\mathcal{M} \models \phi$$

- example: $\phi = \text{temperature always below 37° celsius}$
Model Checking

“does a computing system fulfil its specification?”

- formal model $\mathcal{M}$ of system
- specification $\phi$
- automatic proof or refutation of

$$\mathcal{M} \models \phi$$

- example: $\phi =$ temperature always below $37^\circ$ celsius

initial condition \hspace{2cm} error

here: temperature equal to or above $37^\circ$ celsius
Probabilities

- probabilistic behaviour in system

Thus, cannot always show complete safeness.

Want quantitative bounds on system behaviour.

E.g. "max prob to go above 37° within 20 years: ≤ 10^{-40}".
Probabilities

- probabilistic behaviour in system
  e.g. sensors might fail with given probability

thus, cannot always show complete safeness

want quantitative bounds on system behaviour

e.g. "max prob to go above 37\degree within 20 years: \leq 10^{-40}"

must integrate probabilistic behaviour in system model
Probabilities

- probabilistic behaviour in system
e.g. sensors might fail with given probability

- thus, cannot always show complete safeness
probabilistic behaviour in system
  e.g. sensors might fail with given probability

thus, cannot always show complete safeness

want **quantitative** bounds on system behaviour
  e.g. “max prob to go above 37° within 20 years: $\leq 10^{-40}$”
Probabilities

- probabilistic behaviour in system
e.g. sensors might fail with given probability

thus, cannot always show complete safeness

- want **quantitative** bounds on system behaviour
e.g. “max prob to go above 37° within 20 years: \( \leq 10^{-40} \)”

- must integrate probabilistic behaviour in system model
Analysis of Stochastic Hybrid Systems

- apply **model checking** to stochastic hybrid systems
- explore all states of the model
- combine with property
- apply analysis method to analyse state-transition system

![State Transition Diagram](image)
Analysis of Stochastic Hybrid Systems

- apply **model checking** to stochastic hybrid systems
- explore all states of the model
- combine with property
- apply analysis method to analyse state-transition system

---

- problem: state space **uncountably large**
Analysis of Stochastic Hybrid Systems

- apply **model checking** to stochastic hybrid systems
  explore all states of the model
  combine with property
  apply analysis method to analyse state-transition system

- problem: state space **uncountably large**

- thus, cannot be constructed explicitly
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many \textbf{abstract} states
- apply model checking there
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there

![Diagram of a stochastic hybrid system with transitions labeled by probabilities and actions labeled as 'a' and 'b'.]
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many **abstract** states
- apply model checking there

![Diagram of stochastic hybrid system with abstract states and transitions]
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there

How to construct such a finite model?

Correctness?

Which models can we handle?

And which properties?
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there

![Diagram of stochastic hybrid system with states and transitions labeled with probabilities.]

- How to construct such a finite model?
- Correctness?
- Which models can we handle?
- And which properties?
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many \textit{abstract} states
- apply model checking there

- how to construct such a finite model?
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there

how to construct such a finite model?
- correctness?
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many **abstract** states
- apply model checking there

![Diagram of stochastic system](image)

- how to construct such a finite model?
- correctness?
- which models can we handle?
Abstraction of Stochastic Hybrid Systems

- idea: combine to finitely many abstract states
- apply model checking there

how to construct such a finite model?
- correctness?
- which models can we handle?
- and which properties?
Contribution

- generic framework for general stochastic hybrid systems
Contribution

- generic framework for general stochastic hybrid systems
- provides conservative bounds for properties
Contribution

- generic framework for general stochastic hybrid systems
- provides conservative bounds for properties
- requires no manual intervention
Contribution

- generic framework for general stochastic hybrid systems
- provides conservative bounds for properties
- requires no manual intervention

\[
\text{classical hybrid automata} \quad \downarrow \\
\text{probabilistic hybrid automata}
\]

- builds on classical hybrid solvers (important research area)
Contribution

- generic framework for general stochastic hybrid systems
- provides conservative bounds for properties
- requires no manual intervention

builds on classical hybrid solvers (important research area)
applicable to wide area of models and properties
Classical Hybrid Automata
Hybrid Automata (HA)

\[ \mathcal{H} = (M, \overline{m}) \]

- \( M \): finite set of **modes**
- \( \overline{m} \): **initial mode**

\[ \begin{align*}
\text{Init} & \quad \dot{T} = 2 \land T \leq 10 \land t \leq 3 \\
\text{Heat} & \quad \dot{T} = -T \land T \geq 5 \\
\text{Check} & \quad \dot{T} = -\frac{T}{2} \land t \leq 1 \land t \geq 0.5 \\
\text{Cool} & \quad \dot{T} = -T \land T \geq 6 \land t \geq 2 \land t \geq 0.5 \\
\text{Error} & \quad T \geq 9 \rightarrow T' = T \land t' = 0 \\
\end{align*} \]
Hybrid Automata (HA)

\[ \mathcal{H} = (M, \overline{m}, k) \]

- **\( M \):** finite set of **modes**
- **\( \overline{m} \):** initial mode
- **\( k \):** dimension of the automaton

\[ \begin{align*}
\text{Init} & : \quad \dot{T} = 2 & \quad \text{and} & \quad T \leq 10 \\
\text{Heat} & : \quad \dot{T} = \pi & \quad \text{and} & \quad t \leq 3 \\
\text{Check} & : \quad \dot{T} = -T/2 & \quad \text{and} & \quad t \leq 1 \\
\text{Cool} & : \quad \dot{T} = -T & \quad \text{and} & \quad T \geq 5 \\
\text{Error} & : \quad T \geq 9 \Rightarrow T' = T & \quad \text{and} & \quad t' = 0 \\
\text{true} & : \quad 9 \leq T \leq 10 & \quad \text{and} & \quad t' = 0 \\
\end{align*} \]
Hybrid Automata (HA)

\[ \mathcal{H} = (M, \bar{m}, k, \langle Post_m \rangle_{m \in M}) \]

- \( M \): finite set of modes
- \( \bar{m} \): initial mode
- \( k \): dimension of the automaton
- \( Post_m \): timed behaviour

**Init**

Heat
\[
\begin{align*}
\dot{T} &= 2 \\
T &\leq 10 \\
t &\leq 3
\end{align*}
\]

Cool
\[
\begin{align*}
\dot{T} &= -T \\
T &\geq 5
\end{align*}
\]

Check
\[
\begin{align*}
\dot{T} &= -T/2 \\
t &\leq 1
\end{align*}
\]

Error

\[ t, T, c \]

[10/27] MalerMP91
Hybrid Automata (HA)

\[ \mathcal{H} = (M, \bar{m}, k, \langle Post_m \rangle_{m \in M}, Cmds) \]

- \( M \): finite set of **modes**
- \( \bar{m} \): initial mode
- \( k \): **dimension** of the automaton
- \( Post_m \): timed behaviour
- \( Cmds \): finite set of **guarded commands** \( g \rightarrow u \)

- \( g \): guard
- \( u \): update function

\[ \begin{align*}
  \text{Init} & \quad \text{Heat} & \quad \text{Cool} \\
  \text{true} \rightarrow 9 \leq T' \leq 10 \land t' = 0 \quad \dot{T} = 2 \quad \dot{T} = -T \\
  t \geq 2 \rightarrow T' = T \land t' = 0 \quad \land T \leq 10 \land t \leq 3 \\
  t \geq 0.5 \rightarrow T' = T \land t' = 0 \quad \land T \geq 5 \\
  t \geq 0.5 \rightarrow T' = T \land t' = 0 \\
  t \geq 0.5 \rightarrow T' = T \land t' = 0 \\
\end{align*} \]

[MalerMP91]
Labelled Transition Systems (LTS)

$\mathcal{M} = (S, \bar{s})$

- $S$: set of states
- $\bar{s}$: initial state

Example path: $0 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 0 \rightarrow b \rightarrow 2 \rightarrow ...$

[Keller76]
Labelled Transition Systems (LTS)

\[ M = (S, \overline{s}, Act) \]

- **S**: set of states
- **\( \overline{s} \)**: initial state
- **Act**: actions

A path is a state-action sequence legal by the transition matrix. For example:

- \( 0 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 0 \rightarrow b \rightarrow 2 \rightarrow \ldots \)
Labelled Transition Systems (LTS)

\[ M = (S, \overline{s}, Act, T) \]

- **S**: set of states
- **\( \overline{s} \)**: initial state
- **Act**: actions
- **\( T \)**: transition matrix

Path: state-action sequence legal by \( T \)

\[ 0 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 1 \rightarrow b \rightarrow 2 \rightarrow a, b \]

[Keller76]
Labelled Transition Systems (LTS)

\[ \mathcal{M} = (S, \bar{s}, \text{Act}, T) \]

- \( S \): set of **states**
- \( \bar{s} \): initial state
- \( \text{Act} \): actions
- \( T \): transition matrix

**path**: state-action sequence legal by \( T \)

\[ 0 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 1 \rightarrow a \rightarrow 0 \rightarrow b \rightarrow 2 \rightarrow \ldots \]
Semantics of Hybrid Automata

\[ \text{LTS } [\mathcal{H}] = (S, \bar{s}) \]

- \( S = M \times \mathbb{R}^k \)
- \( \bar{s} = (\overline{m}, 0, \ldots, 0) \)

\[ \text{Heat} \]
\[ \dot{T} = 2 \wedge T \leq 10 \wedge t \leq 3 \]
\[ T \geq 9 \rightarrow T' = T \land t' = 0 \]
\[ t \geq 2 \rightarrow T' = T \land t' = 0 \]
\[ t \geq 0.5 \rightarrow T' = T \land t' = 0 \]

\[ \text{Cool} \]
\[ \dot{T} = -T \wedge T \geq 5 \]
\[ T \leq 6 \rightarrow T' = T \land t' = 0 \]
\[ T \geq 9 \rightarrow T' = T \land t' = 0 \]
\[ t \geq 0.5 \rightarrow T' = T \land t' = 0 \]

\[ \text{Check} \]
\[ \dot{T} = -T/2 \wedge t \leq 1 \]
\[ t \geq 0.5 \rightarrow T' = T \land t' = 0 \]

\[ \text{Error} \]
\[ T \geq 9 \rightarrow T' = T \land t' = 0 \]
\[ t \geq 0.5 \rightarrow T' = T \land t' = 0 \]

Heat, . . .

Check, 0.5, 7, 016, 3.69

Error, . . .
Semantics of Hybrid Automata

\[ \text{LTS } \mathcal{H} = (S, \bar{s}, \text{Act}) \]

- \( S = M \times \mathbb{R}^k \)
- \( \bar{s} = (\bar{m}, 0, \ldots, 0) \)
- \( \text{Act} = \mathbb{R}_{\geq 0} \cup \text{Cmds} \)

\[
\begin{align*}
\dot{T} &= 2 & \text{Heat} & \text{true} \rightarrow 9 \leq T' \leq 10 \land t' = 0 \\
\dot{T} &= \frac{-T}{2} & \text{Check} & t \geq 0.5 \rightarrow T' = T' = 0 \\
\dot{T} &= -T & \text{Cool} & t \geq 0.5 \rightarrow T' = T' = 0 \\
\dot{T} &= \frac{-T}{2} & \text{Error} & t \geq 0.5 \rightarrow T' = T' = 0 \\
\dot{T} &= \frac{-T}{2} & \text{Error} & t \geq 0.5 \rightarrow T' = T' = 0 \\
\end{align*}
\]

Check, 0.5, 7.016, 3.69

Heat, \ldots

Error, \ldots
Semantics of Hybrid Automata

LTS \([\mathcal{H}] = (S, \bar{s}, Act, T)\)

- \(S = M \times \mathbb{R}^k\)
- \(\bar{s} = (m, 0, \ldots, 0)\)
- \(Act = \mathbb{R}_{\geq 0} \cup Cmds\)
- \(T: \) for \(s \in S\) have transitions

\[
\begin{align*}
\dot{T} &= 2, \quad T \leq 10, \quad t \leq 3 \\
\dot{T} &= -T, \quad T \geq 5 \\
\dot{T} &= -T/2, \quad T \leq 1 \quad t \geq 1 \\
\dot{T} &= 0, \quad T \geq 9, \quad t \geq 0.5 \\
\dot{T} &= 0, \quad T \geq 6, \quad t \geq 0.5 \\
\dot{T} &= 0, \quad T \geq 2 \\
\dot{T} &= 0, \quad T \geq 0.5 \\
\end{align*}
\]
Semantics of Hybrid Automata

\[ \text{LTS } [H] = (S, \overline{s}, \text{Act}, \mathcal{T}) \]

- \( S = M \times \mathbb{R}^k \)
- \( \overline{s} = (m, 0, \ldots, 0) \)
- \( \text{Act} = \mathbb{R}_{\geq 0} \cup \text{Cmds} \)
- \( \mathcal{T} \): for \( s \in S \) have transitions from time \( t \) by \( \text{Post}_m(s, t) \)

\[
\begin{align*}
\text{Heat} & : \dot{T} = 2 \land T \leq 10 \land t \leq 3 \\
\text{Cool} & : \dot{T} = -T \land T \geq 5 \\
\text{Check} & : \dot{T} = -T/2 \land t \leq 1 \\
\text{Error} & : \dot{T} = 9 \land t = 0 \\
\text{Init} & : t \geq 2 \rightarrow T' = T \land t' = 0
\end{align*}
\]
Semantics of Hybrid Automata

\[
\mathcal{LTS} \left[ \mathcal{H} \right] = (S, \overline{s}, Act, \mathcal{T})
\]

- \( S = M \times \mathbb{R}^k \)
- \( \overline{s} = (\overline{m}, 0, \ldots, 0) \)
- \( Act = \mathbb{R}_{\geq 0} \cup Cmds \)
- \( \mathcal{T} \): for \( s \in S \) have transitions
  from time \( t \) by \( Post_m(s, t) \)
  from command \( g \rightarrow u \) by \( u(s) \) if \( g \) fulfilled

\[
\begin{align*}
\text{Heat} & : \dot{T} = 2 \land T \leq 10 \land t \leq 3 \\
\text{Cool} & : \dot{T} = -T \land T \geq 5 \\
\text{Check} & : \dot{T} = -T/2 \land t \leq 1 \\
\text{Init} & : \text{true} \rightarrow 9 \leq T' \leq 10 \land t' = 0 \\
\text{t} & = 2 \rightarrow T' = T \land t' = 0 \\
\text{t} & = 0.5 \rightarrow T' = T \land t' = 0 \\
\text{Error} & : \text{true} \rightarrow 9 \leq T' \leq 10 \land t' = 0 \\
\text{t} & = 2 \rightarrow T' = T \land t' = 0 \\
\text{t} & = 0.5 \rightarrow T' = T \land t' = 0 \\
\text{t} & \geq 0.5 \rightarrow T' = T \land t' = 0 \\
\end{align*}
\]
Reachability

Given $H$, does there exist a path reaching an unsafe mode?

\[
\begin{align*}
\text{Init} & : \quad \dot{T} = 2 \\
& \land T \leq 10 \\
& \land t \leq 3 \\
\text{Heat} & : \quad \dot{T} = 2 \\
& \land T \leq 10 \\
& \land t \leq 3 \\
& \quad t \geq 2 \rightarrow T' = T \land t' = 0 \\
& \quad t \geq 0.5 \rightarrow T' = T \land t' = 0 \\
& \quad T \geq 9 \rightarrow T' = T \land t' = 0 \\
\text{Cool} & : \quad \dot{T} = -T \\
& \land T \geq 5 \\
& \quad t \geq 2 \rightarrow T' = T \land t' = 0 \\
& \quad t \geq 0.5 \rightarrow T' = T \land t' = 0 \\
\text{Check} & : \quad \dot{T} = -T/2 \\
& \land t \leq 1 \\
& \quad t \geq 0.5 \rightarrow T' = T \land t' = 0 \\
\text{Error} & : \quad T \geq 9 \\
& \land T' = T \land t' = 0 \\
\text{Error} & : \quad T \leq 6 \\
& \land T' = T \land t' = 0 \\
\text{Error} & : \quad T \leq 9 \\
& \land T' = T \land t' = 0 \\
\text{Error} & : \quad T \leq 6 \\
& \land T' = T \land t' = 0 \\
\text{Error} & : \quad T \leq 9 \\
& \land T' = T \land t' = 0 \\
\end{align*}
\]
Reachability

Given $\mathcal{H}$, does there exist a path reaching an unsafe mode?

(Init, 0, 0, 0) $\rightarrow$ IH $\rightarrow$ (Heat, 0, 9, 0) $\rightarrow$ 0.5 $\rightarrow$ (Heat, 0.5, 10, 0.5) $\rightarrow$ HCo $\rightarrow$ (Cool, 0, 10, 0.5) $\rightarrow$ 0.69 $\rightarrow$ (Cool, 0.69, 5.016, 1.19) $\rightarrow$ CoH $\rightarrow$ (Heat, 0, 5.016, 1.19) $\rightarrow$ 2 $\rightarrow$ (Heat, 2, 9.016, 3.19) $\rightarrow$ HCh $\rightarrow$ (Check, 0, 9.016, 3.19) $\rightarrow$ 0.5 $\rightarrow$ (Check, 0.5, 7.016, 3.69) $\rightarrow$ ChE $\rightarrow$ (Error, 0, 7.016, 3.69)
Abstraction of Hybrid Automata

\[ \mathcal{M} = (A, \bar{z}) \]

- **A**: covering
- **\( \bar{z} \)**: contains initial state

\[ \begin{align*}
\mathcal{M} & = (A, \bar{z}) \\
A & : \text{covering} \\
\bar{z} & : \text{contains initial state}
\end{align*} \]

\[ \text{Init} \]

\[ \text{Heat} \]

\[ \begin{align*}
\bar{z}_1 & : \text{Heat} \\
t \geq 0, & c \geq 0, \\
t \leq c, & T \leq 10
\end{align*} \]

\[ \text{Check} \]

\[ \begin{align*}
\bar{z}_2 & : \text{Check} \\
t \geq 0, & c \geq 2, \\
t \leq c - 2, & T \leq 10
\end{align*} \]

\[ \text{Error} \]

\[ \begin{align*}
\bar{z}_3 & : \text{Error} \\
c \leq 5
\end{align*} \]

\[ \text{Cool} \]

\[ \begin{align*}
\bar{z}_4 & : \text{Cool} \\
t \geq 0, & c \geq 0, \\
t \leq c, & T \leq 10
\end{align*} \]

\[ \text{Heat} \]

\[ \begin{align*}
\bar{z}_5 & : \text{Heat} \\
t \geq 0, & c \geq 2.5, \\
t \leq c - 2.5, & T \leq 10
\end{align*} \]

\[ \text{Check} \]

\[ \begin{align*}
\bar{z}_6 & : \text{Check} \\
t \geq 0, & c \geq 4.5, \\
t \leq c - 4.5, & T \leq 10
\end{align*} \]

\[ \text{Heat} \]

\[ \begin{align*}
\bar{z}_7 & : \text{Heat} \\
t \geq 0, & c \geq 0, \\
t \leq c - 5, & T \leq 10
\end{align*} \]
Abstraction of Hybrid Automata

\[
\text{LTS } \mathcal{M} = (A, \bar{Z}, \{\tau\} \uplus \text{Cmds})
\]

- **A**: covering
- **\(\bar{Z}\)**: contains initial state
- **\(\tau\)**: abstract timed action
- **Cmds**: commands of \(\mathcal{H}\)

\[
\begin{align*}
Z_0 & \quad \text{Init} \\
Z_1 & \quad \text{Heat} \\
& \quad t \geq 0, c \geq 0, \\
& \quad t \leq c, T \leq 10 \\
Z_2 & \quad \text{Check} \\
& \quad t \geq 0, c \geq 2, \\
& \quad t \leq c - 2, T \leq 10 \\
Z_3 & \quad \text{Error} \\
& \quad c \leq 5 \\
Z_4 & \quad \text{Cool} \\
& \quad t \geq 0, c \geq 0, \\
& \quad t \leq c, T \leq 10 \\
Z_5 & \quad \text{Heat} \\
& \quad t \geq 0, c \geq 2.5, \\
& \quad t \leq c - 2.5, T \leq 10 \\
Z_6 & \quad \text{Check} \\
& \quad t \geq 0, c \geq 4.5, \\
& \quad t \leq c - 4.5, T \leq 10 \\
Z_7 & \quad \text{Heat} \\
& \quad t \geq 0, c \geq 0, \\
& \quad t \leq c - 5, T \leq 10
\end{align*}
\]
Abstraction of Hybrid Automata

\[ \mathcal{M} = (A, \bar{Z}, \{\tau\} \uplus \text{Cmds}, \mathcal{T}_{\text{abs}}) \]

- **A**: covering
- \( \bar{Z} \): contains initial state
- \( \tau \): abstract timed action
- **Cmds**: commands of \( \mathcal{H} \)
- **\( \mathcal{T}_{\text{abs}} \)**: transfer transitions to abstraction

\[ \begin{align*}
\mathcal{Z}_0 & \quad \text{Init} \\
\mathcal{Z}_1 & \quad \text{Heat} \quad t \geq 0, \ c \geq 0, \\
& \quad \quad \quad t \leq c, \ T \leq 10 \\
\mathcal{Z}_2 & \quad \text{Check} \quad t \geq 0, \ c \geq 2, \\
& \quad \quad \quad t \leq c - 2, \ T \leq 10 \\
\mathcal{Z}_3 & \quad \text{Error} \quad c \leq 5 \\
\mathcal{Z}_4 & \quad \text{Cool} \quad t \geq 0, \ c \geq 0, \\
& \quad \quad \quad t \leq c, \ T \leq 10 \\
\mathcal{Z}_5 & \quad \text{Heat} \quad t \geq 0, \ c \geq 2.5, \\
& \quad \quad \quad t \leq c - 2.5, \ T \leq 10 \\
\mathcal{Z}_6 & \quad \text{Check} \quad t \geq 0, \ c \geq 4.5, \\
& \quad \quad \quad t \leq c - 4.5, \ T \leq 10 \\
\mathcal{Z}_7 & \quad \text{Heat} \quad t \geq 0, \ c \geq 0, \\
& \quad \quad \quad t \leq c - 5, \ T \leq 10 
\end{align*} \]
Abstraction of Hybrid Automata

\[ \text{LTS } \mathcal{M} = (A, \overline{z}, \{\tau\} \cup \text{Cmds}, T_{\text{abs}}) \]

- **A**: covering
- **\( \overline{z} \)**: contains initial state
- **\( \tau \)**: abstract timed action
- **Cmds**: commands of \( \mathcal{H} \)
- **\( T_{\text{abs}} \)**: transfer transitions to abstraction

\[ z_0 \xrightarrow{\text{IH}} z_1 \]

- **\( z_1 \)**: Heat
  - \( t \geq 0, c \geq 0, t \leq c, T \leq 10 \)

\[ z_1 \xrightarrow{\text{HCh}} z_2 \]

- **\( z_2 \)**: Check
  - \( t \geq 0, c \geq 2, t \leq c-2, T \leq 10 \)

\[ z_2 \xrightarrow{\text{ChE}} z_3 \]

- **\( z_3 \)**: Error
  - \( c \leq 5 \)

\[ z_3 \xrightarrow{\text{ChE}} z_4 \]

- **\( z_4 \)**: Cool
  - \( t \geq 0, c \geq 0, t \leq c, T \leq 10 \)

\[ z_4 \xrightarrow{\text{HCo}} z_5 \]

- **\( z_5 \)**: Heat
  - \( t \geq 0, c \geq 2.5, t \leq c-2.5, T \leq 10 \)

\[ z_5 \xrightarrow{\text{HCh}} z_6 \]

- **\( z_6 \)**: Check
  - \( t \geq 0, c \geq 4.5, t \leq c-4.5, T \leq 10 \)

\[ z_6 \xrightarrow{\text{ChE}} z_7 \]

- **\( z_7 \)**: Heat
  - \( t \geq 0, c \geq 0, t \leq c-5, T \leq 10 \)

- wide tool support exists (HSolver, PHAVer, SpaceEx, etc.)
Correctness of HA Abstraction

- follows from simulation relation semantics → abstraction

[Milner71]
Correctness of HA Abstraction

- follows from simulation relation semantics $\rightarrow$ abstraction
- $R \subseteq S \times A$  
[Milner71]
Correctness of HA Abstraction

- follows from simulation relation semantics $\rightarrow$ abstraction
- $R \subseteq S \times A$
- $\bar{s} R \bar{s}$

[Milner71]
Correctness of HA Abstraction

- follows from simulation relation semantics → abstraction
  \[ R \subseteq S \times A \]

- if \( s R z \),
  for \( a \)-labelled successor in simulated model,
  have \( a \)-labelled successor in simulating model,
  so that the two are also related
Correctness of HA Abstraction

- follows from simulation relation semantics $\rightarrow$ abstraction
  \[ R \subseteq S \times A \]
  \[ \overline{s} R \overline{z} \]
  if $s R z$, for $a$-labelled successor in simulated model, have $a$-labelled successor in simulating model, so that the two are also related

- maintains safeness
Analysis of Stochastic Hybrid Systems
Probabilistic Hybrid Automata (PHA)

\[ \mathcal{H} = (M, \overline{m}, k, \langle Post_m \rangle_{m \in M}) \]

- \( M \): finite set of modes
- \( \overline{m} \): initial mode
- \( k \): dimension of the automaton
- \( Post_m \): timed behaviour

\[ \text{Init} \]

Heat
\[ \dot{T} = 2 \]
\[ T \leq 10 \]
\[ t \leq 3 \]

Cool
\[ \dot{T} = -T \]
\[ T \geq 5 \]

Check
\[ \dot{T} = -T/2 \]
\[ t \leq 1 \]

Error

true
\[ 9 \leq T \leq 10 \]
\[ t' = 0 \]
\[ t \geq 2 \]
\[ T' = T \]
\[ t' = 0 \]
\[ t \geq 0.5 \]
\[ T' = T \]
\[ t' = 0 \]

[Sproston00]
Probabilistic Hybrid Automata (PHA)

\[ \mathcal{H} = (M, \overline{m}, k, \langle Post_m \rangle_{m \in M}, Cmds) \]

- \( M \): finite set of modes
- \( \overline{m} \): initial mode
- \( k \): dimension of the automaton
- \( Post_m \): timed behaviour
- \( Cmds \): finite set of \textit{probabilistic} guarded commands

\[ g \to p_1 : u_1 + \ldots + u_n : u_n \]

\( g \): guard
\( u_i \): updates \hspace{1cm} p_i: \textit{probabilities}
Probabilistic Automata (PA)

\[ \mathcal{M} = (S, \bar{s}, \text{Act} ) \]

- \( S \): set of states
- \( \bar{s} \): initial state
- \( \text{Act} \): actions

\[
\begin{array}{ccc}
0 & \rightarrow & 1 \\
1 & \rightarrow & 2 \\
2 & \rightarrow & 3 \\
3 & \rightarrow & 0
\end{array}
\]

\[ \text{SegalaL95} \]
Probabilistic Automata (PA)

\[ M = (S, s, Act, T) \]

- **S**: set of states
- **\( s \)**: initial state
- **Act**: actions
- **\( T \)**: **probabilistic** transition matrix

0
\[ \begin{array}{ccc}
0 & \rightarrow & b \\
\overset{0.25}{a} & \rightarrow & 0.5 \\
\overset{0.25}{a} & \rightarrow & 0.6 \\
\end{array} \]

1
\[ \begin{array}{ccc}
\overset{a}{0.3} & \rightarrow & b \\
\overset{a}{0.1} & \rightarrow & 0.1 \\
\overset{a}{0.1} & \rightarrow & 0.1 \\
\end{array} \]

2
\[ \begin{array}{ccc}
\overset{a, b}{0.5} & \rightarrow & 0.25 \\
\overset{a, b}{0.6} & \rightarrow & 0.6 \\
\overset{a, b}{0.6} & \rightarrow & 0.6 \\
\end{array} \]

3
\[ \begin{array}{ccc}
\overset{a, b}{0.5} & \rightarrow & 0.25 \\
\overset{a, b}{0.6} & \rightarrow & 0.6 \\
\overset{a, b}{0.6} & \rightarrow & 0.6 \\
\end{array} \]

\[ \text{SegalaL95} \]
Probabilistic Automata (PA)

\( \mathcal{M} = (S, \bar{s}, \text{Act}, \mathcal{T}) \)

- \( S \): set of states
- \( \bar{s} \): initial state
- \( \text{Act} \): actions
- \( \mathcal{T} \): \textit{probabilistic} transition matrix

\begin{align*}
0 & \rightarrow b \rightarrow [0 \rightarrow 0.25, 1 \rightarrow 0.25, 2 \rightarrow 0.5] \rightarrow 2 \rightarrow a \rightarrow [2 \rightarrow 1] \ldots
\end{align*}

[SegalaL95]
Probabilistic Automata (PA)

\[ M = (S, \bar{s}, Act, T) \]
- \( S \): set of states
- \( \bar{s} \): initial state
- \( Act \): actions
- \( T \): \textbf{probabilistic} transition matrix

\begin{align*}
0 & \rightarrow b \rightarrow [0 \rightarrow 0.25, 1 \rightarrow 0.25, 2 \rightarrow 0.5] \rightarrow 2 \rightarrow a \rightarrow [2 \rightarrow 1] \ldots \\
\text{scheduler } \sigma & \in Sched_M: \text{ fixes decisions over successors}
\end{align*}

[SegalaL95]
Probabilistic Automata (PA)

\( \mathcal{M} = (S, \overline{s}, \text{Act}, \mathcal{T}) \)

- \( S \): set of states
- \( \overline{s} \): initial state
- \( \text{Act} \): actions
- \( \mathcal{T} \): probabilistic transition matrix

path: sequence state-action-distribution e.g.
0 → b → [0→0.25, 1→0.25, 2→0.5] → 2 → a → [2→1]...

scheduler \( \sigma \in \text{Sched}_\mathcal{M} \): fixes decisions over successors

induces measure \( Pr_{\mathcal{M},\sigma} \) on sets of paths

[SegalaL95]
Semantics of Probabilistic Hybrid Automata

$$LTS \llbracket \mathcal{H} \rrbracket = (S, \bar{s}, Act)$$

- $$S = M \times \mathbb{R}^k$$
- $$\bar{s} = (\bar{m}, 0, \ldots, 0)$$
- $$Act = \mathbb{R}_{\geq 0} \cup Cmds$$

Heat, ...  

Check, 0.5, 7.016, 3.69

Error, ...
Semantics of Probabilistic Hybrid Automata

\[ \text{LTS } \mathcal{H} = (S, \bar{s}, \text{Act}, \mathcal{T}) \]

- \( S = M \times \mathbb{R}^k \)
- \( \bar{s} = (m, 0, \ldots, 0) \)
- \( \text{Act} = \mathbb{R}_{\geq 0} \cup \text{Cmds} \)
- \( \mathcal{T} \): for \( s \in S \) have transitions from command \( g \rightarrow p_1: u_1 + \ldots + u_n: u_n \) by \( u(s) \) if \( g \) fulfilled from time \( t \) by \( \text{Post}_m(s, t) \)

\[
\begin{align*}
\text{Heat} & : \quad \dot{T} = 2 & T < 10 & \land T \leq 3 \\
\text{Cool} & : \quad \dot{T} = -T & T \geq 5 \\
\text{Check} & : \quad \dot{T} = -T/2 & T \leq 1 \\
\text{Init} & : \quad T = 0 \\
\text{Check, Error} & : \quad T' = T & t' = 0 \\
\text{Heat, Cool, Check} & : \quad T < 9 & T < 6 \\
\text{Error, Ch} & : \quad T > 9 & T > 6 \\
\end{align*}
\]
Probabilistic Reachability

- consideration of single path insufficient
Probabilistic Reachability

- consideration of single path insufficient
- scheduler $\sigma \in Sched_\mathcal{M}$ induces measure on path sets

\[ Pr_{\mathcal{M}, \sigma} \]

Error
Probabilistic Reachability

- consideration of single path insufficient
- scheduler $\sigma \in Sched_M$ induces measure on path sets
- want probability of paths reaching bad state

$$Pr_{M,\sigma}$$

Error
Probabilistic Reachability

- consideration of single path insufficient
- scheduler $\sigma \in Sched_{\mathcal{M}}$ induces measure on path sets
- want probability of paths reaching bad state
- interested in worst case, thus supremum over schedulers

\[ \sup_{\sigma \in Sched_{\mathcal{M}}} Pr_{\mathcal{M},\sigma} \]
Abstraction of Probabilistic Hybrid Automata

\[ \mathcal{M} = (A, \bar{z}, \{\tau\} \cup Cmds, T_{\text{abs}}) \]

- \( A, \bar{z}, \tau, Cmds \): as before

**State 0:** Init

**State 1:** Heat
- \( t \geq 0, c \geq 0, t \leq c, T \leq 10 \)

**State 2:** Check
- \( t \geq 0, c \geq 2, t \leq c - 2, T \leq 10 \)

**State 3:** Error
- \( c \leq 5 \)

**State 4:** Cool
- \( t \geq 0, c \geq 0, t \leq c, T \leq 10 \)

**State 5:** Heat
- \( t \geq 0, c \geq 2.5, t \leq c - 2.5, T \leq 10 \)

**State 6:** Check
- \( t \geq 0, c \geq 4.5, t \leq c - 4.5, T \leq 10 \)

**State 7:** Heat
- \( t \geq 0, c \geq 0, t \leq c - 5, T \leq 10 \)
Abstraction of Probabilistic Hybrid Automata

\[ \mathcal{M} = (A, \overline{z}, \{\tau\} \cup \text{Cmds}, \mathcal{T}_{\text{abs}}) \]

- \( A, \overline{z}, \tau, \text{Cmds} \): as before
- \( \mathcal{T}_{\text{abs}} \): probabilistic

\[
\begin{align*}
\dot{T} & = 2 & \text{Heat} & t \geq 0, \ c \geq 0, \\
& \quad \land t \leq c, \ T \leq 10 \\
\dot{T} & = -1 & \text{Cool} & t \geq 0 \\
& \quad \land t \leq c, \ T \leq 10 \\
\dot{T} & = 0.95 & \text{Error} & c \leq 5 \\
\end{align*}
\]
Abstraction of Probabilistic Hybrid Automata

\[ \mathcal{M} = (A, \bar{A}, \{ \tau \} \cup \text{Cmds}, T_{\text{abs}}) \]

- **A, \bar{A}, \tau, \text{Cmds}:** as before
- **T_{\text{abs}}:** probabilistic

\[ \begin{align*}
\text{Init} & : t \geq 0, c \geq 0, t \leq c, T \leq 10 \\
\text{Cool} & : t \geq 0, c \geq 2.5, t \leq c - 2.5, T \leq 10 \\
\text{Check} & : t \geq 0, c \geq 4.5, t \leq c - 4.5, T \leq 10 \\
\text{Error} & : c \leq 5
\end{align*} \]

- how to obtain such an abstraction?
Constructing Abstractions of PHAs

- consider probabilistic hybrid automaton $\mathcal{H}$
Constructing Abstractions of PHAs

- consider probabilistic hybrid automaton $\mathcal{H}$
- consider non-probabilistic version $\text{ind}(\mathcal{H})$ of $\mathcal{H}$
- replace $c = g \rightarrow p_1 : u_1 + \ldots + p_n : u_n$
- by $\text{ind}(c) = \{g \xrightarrow{\ell_1} u_1, \ldots, g \xrightarrow{\ell_n} u_n\}$
Constructing Abstractions of PHAs

- consider probabilistic hybrid automaton $\mathcal{H}$
- consider non-probabilistic version $\text{ind}(\mathcal{H})$ of $\mathcal{H}$
- replace $c = g \rightarrow p_1 : u_1 + \ldots + p_n : u_n$
  
  by $\text{ind}(c) = \{g \xrightarrow{l_1} u_1, \ldots, g \xrightarrow{l_n} u_n\}$
- consider abstraction $\text{abs}(\text{ind}(\mathcal{H}))$ of $\text{ind}(\mathcal{H})$
Constructing Abstractions of PHAs

- consider probabilistic hybrid automaton $\mathcal{H}$
- consider non-probabilistic version $\text{ind}(\mathcal{H})$ of $\mathcal{H}$
- replace $c = g \rightarrow p_1 : u_1 + \ldots + p_n : u_n$
  by $\text{ind}(c) = \{ g^{\ell_1} u_1, \ldots, g^{\ell_n} u_n \}$
- consider abstraction $\text{abs}(\text{ind}(\mathcal{H}))$ of $\text{ind}(\mathcal{H})$
- use $\text{abs}(\text{ind}(\mathcal{H}))$ to compute abstraction $\text{abs}(\mathcal{H})$ of $\mathcal{H}$
  using labellings $\ell_i$ of $\text{ind}(\text{Cmds})$
Correctness of PHA Abstraction

- follows from simulation relation semantics $\rightarrow$ abstraction

- $R \subseteq S \times A$

- $\overline{s} R \overline{z}$

[SegalaL95]
Correctness of PHA Abstraction

- follows from **simulation relation** semantics → abstraction
- $R \subseteq S \times A$
- $\overline{sRz}$
- if $sRz$,
  for $a$-labelled successor **distribution** in simulated model,
  have $a$-labelled successor distribution in simulating model,
  so that the two are related

[SegalaL95]
Correctness of PHA Abstraction

- follows from simulation relation semantics → abstraction
- \[ R \subseteq S \times A \]
- \( \bar{s} R \bar{z} \)
- if \( s R z \),
  for \( a \)-labelled successor distribution in simulated model,
  have \( a \)-labelled successor distribution in simulating model,
  so that the two are related

[SegalaL95]

- maintains safeness
Rewards

- interest in properties other than reachability
Rewards

- interest in properties other than reachability
- attach rewards

\[
\begin{align*}
\text{Init} & : \quad T = 2 \\
& \quad \wedge T \leq 10 \\
& \quad \wedge t \leq 3 \\
& \quad t \geq 2 \rightarrow T' = T \wedge t' = 0 \\
& \quad t \geq 0.5 \rightarrow \quad 0.05: T' = T \wedge t' = 0 \\
& \quad t \geq 0.5 \rightarrow \quad 0.95: T' = T \wedge t' = 0 \\
\text{Heat} & : \quad \dot{T} = 2 \\
& \quad \wedge T \leq 10 \\
& \quad \wedge t \leq 3 \\
& \quad T \geq 9 \rightarrow T' = T \wedge t' = 0 \\
& \quad T \leq 6 \rightarrow T' = T \wedge t' = 0 \\
\text{Check} & : \quad \dot{T} = -T/2 \\
& \quad \wedge t \leq 1 \\
& \quad t \geq 0.5 \rightarrow \quad 0.05: T' = T \wedge t' = 0 \\
\text{Cool} & : \quad \dot{T} = -T \\
& \quad \wedge T \geq 5 \\
\text{Error} & : \quad t \geq 0.5 \rightarrow \quad T' = T \wedge t' = 0
\end{align*}
\]
Rewards

- interest in properties other than reachability
- attach **rewards**
- obtained per command execution
Rewards

- interest in properties other than reachability
- attach rewards
- obtained per command execution
- or per time unit

**Rewards**

- interest in properties other than reachability
- attach rewards
- obtained per command execution
- or per time unit
Rewards Semantics

- induce two reward structures in PA semantics

![Diagram]

- $\text{Check}, 0.5, 7.016, 3.69$
- $\text{Heat, ...}$
- $0.95$
- $\text{Ch}$
- $0.05$
- $\text{Error, ...}$
Rewards Semantics

- induce two reward structures in PA semantics value \( (\text{rew}_{\text{val}}) \)

\[
\text{Check, 0.5, 7.016, 3.69} \quad \text{Ch} \quad 0.95 \quad \text{rew}_{\text{val}} = 1 \\
\text{Heat, . . .} \quad \text{0.95} \\
\text{Error, . . .} \quad \text{0.05} \\
\text{rew}_{\text{val}} = 3.5
\]
Rewards Semantics

- Induce two reward structures in PA semantics:
  - Value ($rew_{val}$) and time ($rew_{tme}$)

\[
rew_{val} = 3.5, \quad rew_{tme} = 0.5
\]
Rewards Semantics

- induce two reward structures in PA semantics value \((\text{rew}_{\text{val}})\) and time \((\text{rew}_{\text{tme}})\)

\[
\text{Check, 0.5, 7.016, 3.69} \quad \begin{array}{c}
\text{Heat, ...} \\
0.5 \\
\end{array} \\
\text{Ch} \\
\begin{array}{c}
0.95 \\
\text{rew}_{\text{val}} = 1, \text{rew}_{\text{tme}} = 0 \\
0.05 \\
\end{array}
\]

\[
\text{Error, ...} \\
\text{ rew}_{\text{val}} = 3.5, \\
\text{ rew}_{\text{tme}} = 0.5
\]

- can now express properties accumulated:

\[
\text{val}^\sigma_{\mathcal{M}, \text{rew}}, \text{acc} \overset{\text{def}}{=} E_{\mathcal{M}, \sigma} \left[ \lim_{n \to \infty} \sum_{i=0}^{n} \text{rew}_{\text{val}} \right]
\]

long-run:

\[
\text{val}^\sigma_{\mathcal{M}, \text{rew}}, \text{lra} \overset{\text{def}}{=} E_{\mathcal{M}, \sigma} \left[ \lim_{n \to \infty} \frac{\sum_{i=0}^{n} \text{rew}_{\text{val}}}{\sum_{i=0}^{n} \text{rew}_{\text{tme}}} \right]
\]
Rewards Semantics

- Induce two reward structures in PA semantics: value ($rew_{val}$) and time ($rew_{tme}$).

  - $rew_{val} = 3.5, 7.016, 3.69$  
  - $rew_{tme} = 0.5$

  - $rew_{val} = 1, rew_{tme} = 0$
  - $rew_{val} = 0.5$
  - $rew_{val} = 0.05$

- Can now express properties accumulated:
  - $val_{M, rew, acc}^\sigma \overset{\text{def}}{=} E_{M, \sigma} \left[ \lim_{n \to \infty} \sum_{i=0}^{n} rew_{val} \right]$
  - Long-run: $val_{M, rew, lra}^\sigma \overset{\text{def}}{=} E_{M, \sigma} \left[ \lim_{n \to \infty} \frac{\sum_{i=0}^{n} rew_{val}}{\sum_{i=0}^{n} rew_{tme}} \right]$

- Interested in min/max of values.
Abstraction for Rewards

transform timed rewards → command rewards

\[
\begin{align*}
Z_0 & \text{ Init} \\
Z_1 & \text{ Heat } t \geq 0, T \leq 10 \\
Z_2 & \text{ Check } t \geq 0, T \leq 10 \\
Z_3 & \text{ Error} \\
Z_4 & \text{ Cool } t \geq 0, T \leq 10 \\
Z_5 & \text{ Heat } t \geq 0, T \leq 10 \\
Z_6 & \text{ Check } t \geq 0, T \leq 10 \\
Z_7 & \text{ Heat } t \geq 0, T \leq 10
\end{align*}
\]
Abstraction for Rewards

- transform timed rewards $\rightarrow$ command rewards
- compute reward structures for abstraction

$\sigma := \text{initial scheduler}$

$\text{repeat}
\begin{align*}
\text{compute reachability probability } v \text{ under } \sigma \\
\text{forall the } z \in A \text{ do}
\text{improve } \sigma(z) \text{ if possible}
\end{align*}
\text{until no further improvement possible}$

$\text{return } (v, \sigma)$
Abstraction for Rewards

- transform timed rewards $\rightarrow$ command rewards
- compute reward structures for abstraction
- compute overapproximation of values

```
1 σ := initial scheduler
2 repeat
3    compute reachability probability v under σ
4   forall the z ∈ A do
5       improve σ(z) if possible
6 until no further improvement possible
7 return (v, σ)
```
Abstraction for Rewards

- transform timed rewards $\rightarrow$ command rewards
- compute reward structures for abstraction
- compute overapproximation of values
- correctness: extended simulation relation

$z_0$ Init

$z_1$ Heat $t \geq 0, T \leq 10$

$z_2$ Check $t \geq 0, T \leq 10$

$z_3$ Error

$z_4$ Cool $t \geq 0, T \leq 10$

$z_5$ Heat $t \geq 0, T \leq 10$

$z_6$ Check $t \geq 0, T \leq 10$

$z_7$ Heat $t \geq 0, T \leq 10$

$\sigma :=$ initial scheduler

repeat

1. $\sigma :=$ initial scheduler
2. repeat
3. compute reachability probability $v$ under $\sigma$
4. forall the $z \in A$
5. improve $\sigma(z)$ if possible
6. until no further improvement possible
7. return $(v, \sigma)$
Computing Reward Structures

- depends on hybrid automata tool

\[
\begin{align*}
Z_3 &\quad \text{Error} \\
2t \geq T \\
\wedge 2t - 5 &\leq T \\
\wedge 2T &\geq 5 - t \\
\wedge 2T &\leq 10 - t
\end{align*}
\]

max time stay in error?
Computing Reward Structures

- depends on hybrid automata tool
- for polyhedra: maps to linear programming

\[ z_3 \]

Error

\[
\begin{align*}
2t & \geq T \\
\land 2t - 5 & \leq T \\
\land 2T & \geq 5 - t \\
\land 2T & \leq 10 - t
\end{align*}
\]

max time stay in error?

\[ \text{rew val} = 4, \text{rew tme} = 1 \]
Computing Reward Structures

- depends on hybrid automata tool
- for polyhedra: maps to linear programming

\[
\begin{align*}
Z_3 & \quad \text{Error} \\
2t & \geq T \\
\land 2t - 5 & \leq T \\
\land 2T & \geq 5 - t \\
\land 2T & \leq 10 - t
\end{align*}
\]

max time stay in error?

\[
\begin{align*}
\text{max time stay in error?} &= 4 \\
\text{rew tme} &= 1
\end{align*}
\]
Computing Reward Structures

- depends on hybrid automata tool
- for polyhedra: maps to linear programming

rew_{val} = 4, rew_{tme} = 1

\[ 2t \geq T \land 2t - 5 \leq T \land 2T \geq 5 - t \land 2T \leq 10 - t \]

max time stay in error?

max / min \( t \Rightarrow t \in [1, 4] \)
Abstraction enables automatic verification of a very general class of properties of generic stochastic hybrid automata, by extending existing established methods.

- Classical hybrid automata
- Probabilistic hybrid automata
  - Continuous distributions
  - Partial control
  - Parameters
  - Rewards
  - Orthogonal combinations